An Improved Item-based Collaborative Filtering Recommendation System

Lan-jun YAO\(^1\)*, Li-hong SHANG\(^1\) and Mi ZHOU\(^2\)

\(^1\) School of Computer Science and Engineering, Beihang University, Beijing 100191, China
\(^2\) Reliable Avionics technology Co. Ltd., Haidian District Beijing 100191, China

* zy1406125@buaa.edu.cn
*Corresponding author

Keywords: Item-based collaborative filtering, Similarity transmit, Trust network.

Abstract. Today E-commerce is very popular, recommendation systems are very widely applied to various sites [1]. However, there still remains many problems in current recommendation system to be resolved. Given the data sparse problem in the traditional collaborative filter algorithm, we will introduce the relationship of the trust between the items, and transmit the similarity throughout it. In this paper, Experiment shows that the accuracy and coverage rate of the algorithm have been improved.

Introduction

With the rapid development of the technology and the Internet, people enter the age when information overloads. People have to face massive information every day, so how to find the information that he is interested in or is useful for him in a short time is becoming more and more difficult. But sometimes users do not know what they really need or cannot express their intentions unequivocally. And the electricity providers often require mine the potential interest of customers to attract customers, which can provide greater economic benefits for them.

The most effective way to solve the above problems is to introduce the recommendation system. The so-called recommendation system is a system that by collecting the user information, historical behavior, and other information of items tap the potential items that users are interested in, and recommend the items to the users. The implementation process of the recommendation system is divided to three parts [2]: (1) Collecting the users’ behavior information, including users’ ratings, tour and others acts. (2) Analyzing the preference of the users. Model the users’ behavior and analyze their preference by using the collected information. (3) Predicting the ratings. Predict the ratings for items by the users’ preference, and recommend the top-N items to users. The core of recommendation system is the recommendation algorithm, which directly determines the performance of the recommendation system. According to the different recommendation algorithms, we can divide the recommendation systems into the content-based recommendation system [3], Rule-based recommendation system [4], knowledge-based recommendation system [5], and collaborative filtering recommendation system [6]. The collaborative filtering system is the most successful and widely used recommendation. Although the recommendation system has made some series of the achievement, there are still many problems in practical applications, such as data sparsity, and accuracy problems, cold start, scalability issues, etc. This paper mainly focuses the problem of data sparsity.

Item-based Collaborative Filtering Algorithm

Collaborative filtering recommendation algorithm can be divided into user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. User-based collaborative filtering algorithm computes the similarity among users, predicts the ratings of the items that person hasn’t rated and recommends users’ favorite items to target users by similarity among users. Item-based collaborative filtering recommendation is calculated based on the similarity among items,
computes the ratings of users and recommends the item to users by similarity between items. In general, the method of similarity calculation has three categories:

1. Pearson
   Given a matrix of users and ratings, the Pearson similarity of the item “i” and item “j” is the following:
   \[
   \text{sim}(i, j) = \frac{\sum_{u \in U} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{uj} - \bar{r}_j)^2}}. 
   \]  
   Among them, \( \text{sim}(i, j) \) represents the similarity between item “i” and item “j”, \( U \) represents a set of users, \( r_{ui} \) indicates the rating of user \( i \) for item \( j \), \( \bar{r}_i \) represents the average score of item “i”.

2. Cosine similarity
   Express the rating of item “a” and item “b” by vectors. The formula is as follows:
   \[
   \text{sim}(a, b) = \cos(\bar{a}, \bar{b}) = \frac{\bar{a} \cdot \bar{b}}{|\bar{a}| |\bar{b}|}. 
   \]  
   Among them, the symbol represents the dot product between vectors, \( |\bar{a}| \) represents the European length of representation vector, also the square root of his own dot product.

3. Improved cosine similarity
   The similarity is between 0 and 1, and the value is greater so that the degree of similarity is higher. Basic cosine similarity does not consider the differences in users’ mean ratings. The improved cosine similarity can settle the problems like Pearson whose value between -1 and 1. The improved cosine similarity is calculated as follows:
   \[
   \text{sim}(a, b) = \frac{\sum_{u \in U} (r_{ua} - \bar{r}_a)(r_{ub} - \bar{r}_b)}{\sqrt{\sum_{u \in U} (r_{ua} - \bar{r}_a)^2} \sqrt{\sum_{u \in U} (r_{ub} - \bar{r}_b)^2}}. 
   \]  
   After the similarity between the various items is determined, you can predict the users’ rating of the items, the rating of user \( u \) for item \( p \) is:
   \[
   \text{pred}(u, p) = \frac{\sum_{i \in \mathcal{I}} \text{sim}(i, p) \times r_{ui}}{\sum_{i \in \mathcal{I}} \text{sim}(i, p)}.
   \]  

**Improved item-based Collaborative Filtering Algorithm**

The performance of the item-based collaborative filtering algorithm will reduce when the data is sparse, but the user scoring matrix is often very sparse. In recent years, collaborative filtering algorithms that are integrated into trust networks are emerging, which alleviates the data sparsity problem to a certain extent and improves the performance of the proposed algorithm. In today’s life, there is a trust relationship among users [7], and the users tend to accept the items that those people who they trust like, which can be recommended by a credit user. The collaborative filtering algorithm, which is integrated into the trust network, transmits this kind of trust network and then forecasts the user information.

Based on this trust network transmission concept, we classify similarity into direct similarity and indirect similarity. The direct similarity is calculated by traditional cosine similarity, and the indirect similarity is used to transfer the similarity by the relationship between items. Later, the two similarities were linearly combined to predict the ratings.

When calculating the indirect similarity, we cannot simply transfer it by similarity. In other words, if the similarity between item “a” and item “b” is very high, we cannot think the similarity of item “b” to another item is very high when the direct similarity between item "a" and another item is high. Because the similarity of the transmission process and user-based trust network is similar, whether the similarity of one item to another item is similar to the similarity of another item to another item represents whether an item is trustworthy to another item. We should not only consider the difference between the scoring of items, but also the number of users who score for the item together. Therefore, we model the trust relationships between items.
Because the user scores for two projects at the same time, the smaller difference in scoring between projects, the closer the similarity of one project to the other project for the similarity of the other project to other projects, that is, the target project will be the trust of the project.

We can express the measure of this difference as follows:

\[
rate(i, j) = \frac{success(i, j)}{item(u, i) \cap item(u, j)}.
\]

Herein, \(item(u, i) \cup item(u, j)\) represents the number of users who simultaneously score item “i” and “j”, success \( (i, j) \) indicates that the scores from use “u” for the item “i” and item “j” are less than the threshold.

The algorithm is mainly to solve the problem caused by data sparsity. For a sparse user scoring matrix, if all users score too little on an item, the similarity between the item and the other items will be inaccurate in the similarity calculation process. In this case, in the process of similarity transfer, we should reduce the impact of the scoring too few items on other items, that is to reduce the weight of the similarity calculation process. However, considering that if the weight is directly proportional to the number of items scored by the user, it will lead to the greater impact of the project with a larger number of scoring, so the impact weight is defined as follows:

\[
f(i) = \begin{cases} 1 & \text{(User(i) } \geq \text{ average } \times 0.5) \\ \frac{\text{User(i) - average}}{0.5} & \text{(User(i) } < \text{ average } \times 0.5) \end{cases}
\]

Therefore, in the process of transferring similarity, the weight of on item to another can be expressed as:

\[
T_{i \rightarrow j} = rate(i, j) \times f(j).
\]

So we can define indirect similarity as follow:

\[
\text{indirect \_sim}(i, j) = \frac{\sum_p T_{i \rightarrow p} \times \text{sum}(p, j)}{\sum_p T_{i \rightarrow p}}.
\]

**Experiment Results**

In this paper, we conduct the experiment by using the data from Movielens, the ratings ranging from 1 to 5. The algorithm is mainly to solve the data sparsity problem. In order to test the effect of the algorithm, we randomly extract the sparse degree of 98.6%, 97.6 and 95.5% of data as test data set.

Algorithm evaluation method:

1. Evaluation of the prediction score

In this paper, we use RMSE [8] to evaluate the prediction accuracy of the algorithm. The formula is as follows:

\[
\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(p_i - r_i)^2}{N}}.
\]

2. Score coverage of different sparse degree

As the scoring matrix is too sparse, scoring coverage is also an important indicator of the recommended system. The score coverage refers to the percentage of the predicted score that the recommendation system can predict the user’s score for the item.

\[
\text{RC} = \frac{\text{the count of the items that can be predicted}}{\text{the count of the items that needs to be predicted}}.
\]

In this experiment, we pass on the direct phase degree through the trust value, and select the number of different trust items with different RMSE value. With the number of different trust items, the experimental results obtained RMSE are shown in Figure 1 and Figure 2:
It is found that RMSE decreases with the increase of the number of trust items. However, when it is below a certain value, RMSE gradually become stable or even increases.

The collaborative filtering algorithm (TICF) based on trust similarity transfer mentioned in this paper is compared with item-based collaborative filtering (ICF), user-based collaborative filtering (UCF), and item-based with improved cosine similarity (ICICF). At different sparse levels, the coverage of the predictive scoring is shown in Table 1:

<table>
<thead>
<tr>
<th>sparse</th>
<th>UCF</th>
<th>ICF</th>
<th>ICICF</th>
<th>TICF</th>
</tr>
</thead>
<tbody>
<tr>
<td>97.6%</td>
<td>78.64%</td>
<td>85.36%</td>
<td>85.36%</td>
<td>89.72%</td>
</tr>
<tr>
<td>98.6%</td>
<td>72.57%</td>
<td>79.59%</td>
<td>79.59%</td>
<td>84.69</td>
</tr>
<tr>
<td>99.2%</td>
<td>70.07%</td>
<td>76.63%</td>
<td>76.63%</td>
<td>80.38%</td>
</tr>
</tbody>
</table>

The RMSE with the data set of different sparsity is shown in Table 2:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UCF</th>
<th>ICF</th>
<th>ICICF</th>
<th>TICF</th>
</tr>
</thead>
<tbody>
<tr>
<td>97.6%</td>
<td>1.10429</td>
<td>0.9686</td>
<td>0.9674</td>
<td>0.9639</td>
</tr>
<tr>
<td>98.6%</td>
<td>1.17384</td>
<td>0.9767</td>
<td>0.9732</td>
<td>0.9713</td>
</tr>
<tr>
<td>99.2%</td>
<td>1.24710</td>
<td>1.0916</td>
<td>1.0892</td>
<td>1.0883</td>
</tr>
</tbody>
</table>

Item-based algorithm has improved, but the coverage is lower than the LFM algorithm. In terms of accuracy, TICP algorithm is superior to other algorithms. Therefore, collaborative filtering algorithm based on the trusty and similarity has a certain improvement in accuracy and coverage, which has greatly improved some problems caused by the data sparseness.

**Literature References**

Data sparsity problem has brought many adverse effects to the recommendation algorithm. There are many methods to solve this problem. Luo [9] proposed a local user similarity and global user similarity concept, then predict the ratings according to local nearest local neighbor and global neighbor; Zhang [11] set an iterative threshold to fill in the none scoring items by using the nearest neighbor recursively; Zheng [12] considered the influence of the tag and time information on the
result of recommendation system, and used tag weight and time weight to modify the preference matrix, then calculated the value of preference.

Conclusions
Currently, the recommendation system is very hot. Almost all of the electronic commerce system, are applying recommendation system to their production environment, such as music systems, video systems and so on, and they have received huge economic benefits. The collaborative filtering recommendation system is currently the most widely used and most successful personalized recommendation system. However, there are still some problems in collaborative filtering algorithm, such as cold start, data sparsity and long tail, to be solved. In this paper, based on the traditional collaborative filtering algorithm we classify the similarity into indirect similarity and indirect similarity, then the indirect similarity is computed by transferring the direct similarity through the trust network, so that accuracy and coverage of the improved collaborative filtering has been improved. However, when the matrix of users and ratings is too sparse, the accuracy of the algorithm will be decreased to some extent. Taking into account every recommendation algorithm has both advantages and disadvantages, a recommendation is often a combination of the multiple recommendation algorithms, and in the future I will integrate into other recommendations algorithm to improve the performance.

Acknowledgement
I would like to express my deepest gratitude to my teachers who gave me the criticism and suggestion during the writing of this thesis. And I also appreciate my friends who gave much help to me.

References